Clickbait Post Detection using NLP for Sustainable Content

Ganapati Raju N. V.[1*], *Nikhil* Nyalakanti[1], *Premsai* Kambampati[1], *Yeshwanth* Kanthali[1] *Shivam Pandey*[2], *K.* Maithili[3]

[1]Department of Information Technology, Gokaraju Rangaraju Institute of Engineering and Technology, India

[2]School of Applied and Life Sciences, Uttaranchal University, Dehradun, 248007, India

[3]KG Reddy College of Engineering & Technology, India

Abstract. Clickbait is a significant problem on online media platforms. It misleads users and manipulates their engagement. A user who clicks on a clickbait link may be taken to a website full of ads, or that requires them to pay for something. The goal of this project is to create a system that can recognize clickbait posts so that user can access only to sustainable content. The system will analyze data using natural language processing (NLP) and machine learning techniques. NLP pre-processing techniques, such as tokenization, lemmatization, and stemming, will be utilized to extract essential elements from the headlines. These features will subsequently be used to train a machine learning model, specifically a supervised classifier, to distinguish between clickbait and non-clickbait news headlines. The project will explore a range of algorithms and techniques, including popular text representation models such as TF-IDF or word embeddings, as well as classifier models like logistic regression or random forests. The model will be evaluated using a variety of metrics, such as Accuracy, Precision, Recall, and F1 score. By making it easier for users to identify clickbait, the system can help to reduce the amount of time and money wasted on this type of content.

1 Introduction

Online content known as "clickbait" is made to grab readers' attention and encourage clicks, frequently at the price of factual and trustworthy information. In addition to being sensationalized or deceptive, clickbait headlines often make promises that the content cannot keep. On several websites, including news websites, social media platforms, and blogs, clickbait can be found. Several factors make clickbait problematic. First off, it might jeopardize the authority of online media. Users who click on click-bait headlines are often let down when the content is not what they were hoping for. Second, it can make users' experiences unpleasant. Third, it can be used to disseminate propaganda and false information. There are several strategies for avoiding clickbait. One way is to be aware of clickbait's telltale symptoms. Clickbait headlines frequently employ sensationalized language, make promises the material cannot keep, and create emotional appeals. It's usually true if a headline looks too good to be true. Using browser add-ons might assist you in identifying and avoiding clickbait is another strategy to tackle it. Reporting clickbait to

* Corresponding author: nvgraju@griet.ac.in

the website where it is published can aid in the fight against it. A clickbait headline can be written to the administrators or moderators of the website. Reporting it will help in increasing awareness of the issue and discourage websites from using clickbait in their content. Although clickbait is a severe issue, it can be solved. We can contribute to the internet being a more dependable, sustainable and trustworthy place by being aware of the telltale indications of clickbait, using browser extensions, and reporting websites that use it.

Fig. 1. Click Bait

2 Literature Survey

Clickbait headlines are exaggerated headlines whose main motive is to mislead the reader to "click" on them. They create a nuisance in the online experience by creating a lure towards poor content. This paper is based on convolutional neural networks (CNNs) and recurrent neural networks to propose a deep learning-based strategy for clickbait identification (RNNs) [1]. The authors compare different architectures and evaluate the performance on a large dataset of news headlines, achieving high accuracy in clickbait detection. Lee D. presents other feature engineering techniques for clickbait detection. This research proposes a set of linguistic and structural features extracted from news headlines, including headline length, sentiment analysis, and part-of-speech tags [2]. The effectiveness of these features is evaluated using machine learning algorithms, demonstrating promising results.

Clickbait poses challenges in maintaining the quality of information and user experience on online platforms, effective clickbait detection mechanisms are crucial. This paper presents a hybrid approach combining linguistic features, topic modelling, and deep learning models for clickbait detection of social media headlines [3]. The authors leverage the unique characteristics of social media data and propose a multi-stage classification framework to improve the accuracy of clickbait detection. Clickbait, defined as sensationalized or deceptive information, has become a serious concern of, severe concern in online media platforms.

Detecting clickbait is critical for maintaining the legitimacy of content and user trust. This study introduces an ensemble learning approach for clickbait detection by combining multiple classifiers and incorporating domain knowledge [4]. The authors utilize domain-specific features and develop an ensemble model to capture the underlying patterns in clickbait headlines. The experimental findings show that the proposed method is an effective approach. This research compares different deep neural network topologies for clickbait detection [5]. Authors compare the performance of CNNs, RNNs, and transformer-based models, on a large-scale dataset of news headlines. The findings highlight the strengths and limitations of different architectures in clickbait detection tasks.

Wang J. proposes a novel approach for clickbait detection based on graph-based text representation [6]. The authors construct a graph representation of news headlines by considering words as nodes and their co-occurrence relationships as edges. Later, they employ graph embedding techniques to capture the semantic meaning and context of clickbait headlines. The experimental results show that the proposed method accurately detects clickbait posts [7-20].

3 Methodology

3.1 Dataset

The Dataset was collected from the below Github repository:
https://github.com/bhargaviparanjape/clickbait/tree/master/dataset

The directory contains two files, each consisting of the headlines of 16,000 articles. Both files are compressed using gzip, and each line in the decompressed files contains one article headline.

The clickbait corpus consists of article headlines from BuzzFeed, Upworthy, ViralNova, Thatscoop, Scoopwhoop, and ViralStories. The non-clickbait article headlines are collected from WikiNews, New York Times, The Guardian, and The Hindu.[7]

Table. 1. Dataset

Dataset	Total files
All files	32519
Clickbait	16093
Not_Clickbait	16426

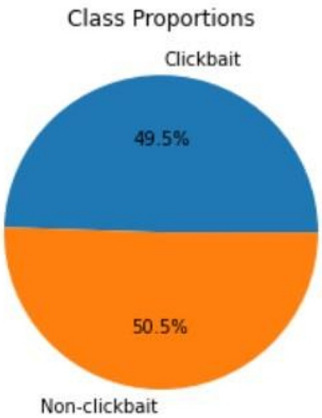

Fig. 2. Dataset proportions

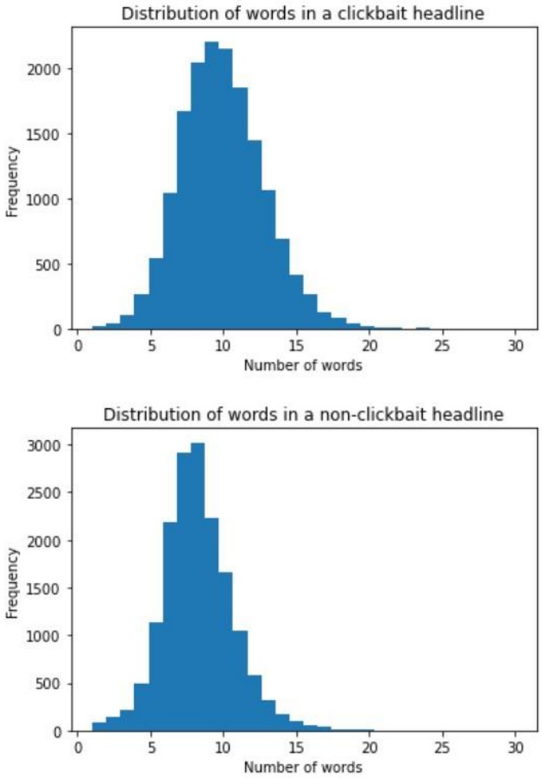

Fig. 3. Word count in both kinds of headlines

3.2 Preprocessing:

The implementation of this system was done in Python Programming Language, which has a rich ecosystem of libraries and is easy to use and read. The following functions were used to preprocess the text data derived from the dataset.

3.2.1 re.sub() function:

This function is based on regular expressions. It is used to replace all the characters which are not English alphabets with spaces. It can be termed a data-cleaning function.

3.2.2 str.lower() function:

This function converts all the letters in a string into lowercase letters. This function makes the text case insensitive toward text comparison and search operations.

3.2.3 str.split() function:

This function is used to split a string into a list. Hence, using this function, the sentences in the dataset are divided into words.

3.2.4 lemmatize() function:

Lemmatization helps normalize different word forms to a single base or origin form. Python's NLTK (Natural Language Toolkit) library consists of WordNetLemmatizer class which provides this function to implement lemmatization.

3.3 NLP Techniques:

3.3.1 TF-IDF Vectorization:

The term frequency-inverse document frequency is abbreviated as TF-IDF. It is a statistical measure that determines the importance of a word in a sentence collection of texts. Simply said, it aids in identifying the most significant words in a document.

TF-IDF Vectorization converts a collection of raw documents into a matrix of TF-IDF features. It combines the functionality of both tokenization and TF-IDF transformation.

TF-IDF approach is widely utilized in information retrieval and text mining. It aids in determining the significance of a word in a document by comparing its frequency in the complete collection of texts. The greater the TF-IDF value, the greater the importance of the word in the paper.

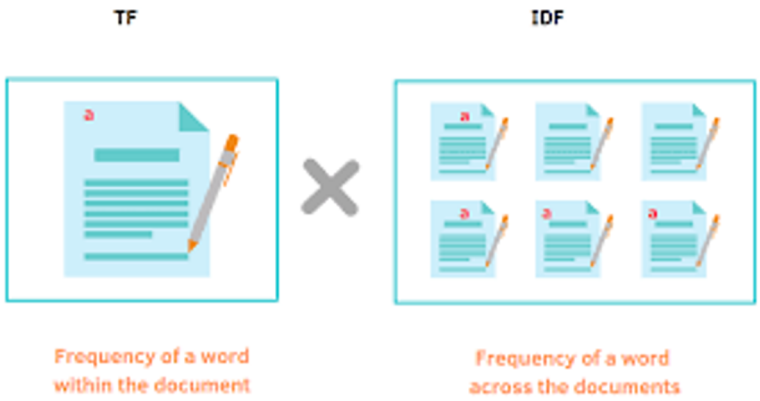

Fig. 4. TF-IDF

3.3.2 Word2Vec:

Word2Vec is a natural language processing technique for generating word embeddings. It is a neural network-based technique that maps words to a high-dimensional vector space and clusters words with similar meanings. This method is used to analyze massive datasets and uncover patterns in word relationships.

By encoding words as numerical vectors, mathematical operations such as addition and subtraction are performed on them to build new word vectors that reflect the underlying meaning of the words.

3.4 Machine Learning Algorithms:

3.4.1 Logistic regression

Logistic regression is a statistical technique for predicting the future analyzing data, and making predictions based on variable relationships. When the dependent variable is a binary variable or categorical, this type of regression analysis is known as used. In simplest terms, it identifies correlations between variables and predicts outcomes based on those associations.

3.4.2 Support vector Classifier:

A Support vector Classifier is a type of machine learning technology used to perform classification problems. It operates by determining the best potential decision boundary between various classes of data items. In layman's words, it assists in identifying patterns in data that distinguish different types from one another and then employs those patterns to produce predictions about fresh data points. The approach is especially beneficial when the data cannot be split into multiple classes using a straight line and is not linearly separable.

3.4.3 Gradient Boosting Classifier:

Gradient Boosting is an approach to machine learning that combines decision trees and Gradient Boosting to increase prediction accuracy. It works by constructing a series of decision trees., each of which attempts to fix the previous tree's faults. In layman's words, it aids in identifying patterns in data that are connected with various outcomes and then employs those patterns to generate predictions about fresh data points. Because it can find essential features and use them to produce accurate predictions, the method is especially beneficial when working with complex datasets with multiple attributes.

3.4.4 Random Forest Classifier:

A Random Forest Classifier is a machine-learning method that predicts using numerous decision trees. It uses many decision trees to train on a subset of the data and a subset of the features. The program then combines the predictions of the different trees to arrive at a final prediction. In layman's words, it aids in identifying patterns in data that are connected with various outcomes and then employs those patterns to generate predictions about fresh data points. Because it can find essential features and use them to produce accurate predictions, the method is especially beneficial when working with complex datasets with multiple attributes.

3.4.5 Voting Classifier

The Voting Classifier is an example of an ensemble learning strategy that integrates results from many machine learning algorithms to increase prediction accuracy. It operates by training multiple machine learning algorithms on the same dataset and merges their predictions. In layman's words, it aids in identifying patterns in data that are connected with various outcomes and then employs those patterns to generate predictions about fresh data points. Because it can find essential features and use them to produce accurate predictions, the method is especially beneficial when working with complex datasets with multiple attributes.

3.5 Workflow and Algorithm:

1. The dataset files are in the form of '.gz.' They were converted to .txt files and then opened in Jupyter Notebook.
2. The files are tokenized into sentences using package punkt of the NLTK library.
3. Now these sentences are pre-processed.
4. The pre-processing involves:
    - Replacing all non-alphabetic characters with spaces
    - lower the cases of all letters in the sentence
    - splitting the strings in each sentence
    - Applying lemmatization on each String(word)
    - Grouping the words into a String again
5. These sentences are then stored in two data frames using pandas, one data frame for clickbait headlines and another for not_clickbait headlines.
6. The data frames consist of two columns:
    - Post_Headline – The sentence column
    - Whether_Clickbait - Class label column
7. The data frames are then concatenated, and the resultant data frame is shuffled for the benefit of using the NLP model effectively.
8. Save the Dataset in an Excel sheet using pandas for further usage.
9. There have been two NLP techniques implemented in this project, they are:
    - TF-IDF
    - Word2Vec
10. *(a) TF-IDF:*
    1. Initially, the pre-processed data is implemented in TF-IDF vectorization.
    2. Every tuple in the Dataset is transformed into a TF-IDF vectorizer object. Then it is converted into a sparse TF-IDF feature matrix. This matrix serves as the input for the model.
    3. The vector data is then fitted into various classification algorithms. They are:
        - Logistic Regression
        - Support Vector Classifier
        - Gradient Boosting classifier
        - Random Forest Classifier
        - Voting classifier

    (b) Word2Vec:
    1. After TF-IDF, the data is re-implemented in the Word2Vec process.
    2. Word2Vec is a popular method of converting words into unique vectors, i.e., generating word embeddings.
    3. The similarity between two words can also be identified using the Word2Vec process.
    4. In this process, a new column in the data frame for tokens is initially created, i.e., every tuple in column "Post_Headline" is divided into words.
    5. Now these words are converted into vectors. The Arithmetic Mean of all word embeddings of each tuple/sentence is taken, and the resultant is stored in a list.
    6. This list is then combined with the class label column to be fitted into a machine-learning algorithm.

4 Result Analysis

Five popular Machine Learning algorithms have been implemented to distinguish between clickbait and non-clickbait headlines. Logistic Regression, Support Vector Classifier (SVC), Random Forest Classifier, Gradient Boosting Classifier, and Voting Classifier. These algorithms have been trained on labeled data, where the headlines are associated with their respective clickbait or non-clickbait labels.
Once the models are trained, they can be used to predict the clickbait likelihood of new headlines. The trained models make predictions based on these vectors, providing a reliable indication of whether a given headline is clickbait.
The results of the project 'Clickbait Post Detection using NLP' are the metrics obtained by training and testing the above machine learning models.
The metrics used in this project are Accuracy, Precision, Recall, F1-score, and Confusion Matrix.
The Machine Learning Algorithms were applied in two methods: TF-IDF and Word2Vec.

4.1 Method-1: TF-IDF:-

The pre-processed data obtained from the TF-IDF vectorizer is fitted into machine learning algorithms. Then the trained models are tested and the following metrics were obtained as follows:

4.1.1 Accuracy

A machine learning model's Accuracy measures how well it can predict the outcomes of new data points. It is calculated by the total number of projections divided by the number of correct predictions.

The graph **Fig. 5.** represents the Accuracy of the five machine learning algorithms. It can be observed that the **Voting Classifier** has the highest Accuracy.

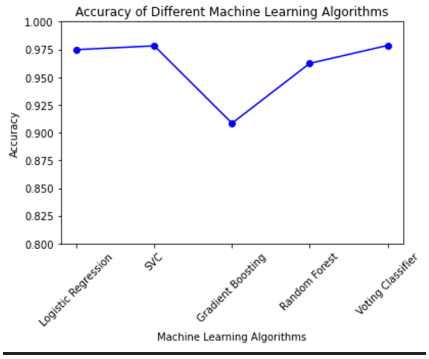

Fig. 5. Accuracy

4.1.2 Precision

Precision is the accuracy with which a machine learning model can distinguish the optimistic outcomes of new data points. It is determined by splitting the overall by dividing the total number of true positives by the number of positive predictions.

The graph **Fig. 6.** represents the Precision of the five machine-learning algorithms. It can be observed that the **Support Vector classifier** has the highest Precision.

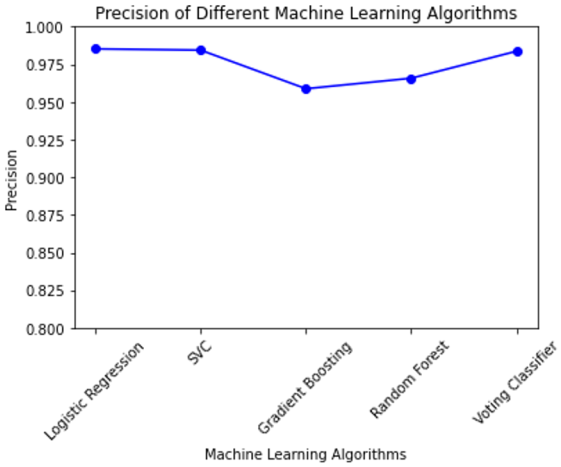

Fig. 6. Precision

4.1.3 Recall

It is a metric that measures how well a machine-learning model recognizes the positive outcomes of new data points. It is derived by dividing the total number of genuine positive events by the total number of true positive forecasts.

The graph **Fig. 7.** represents the Recall of the five machine learning algorithms. It can be observed that the **Support Vector Classifier** has the highest Recall.

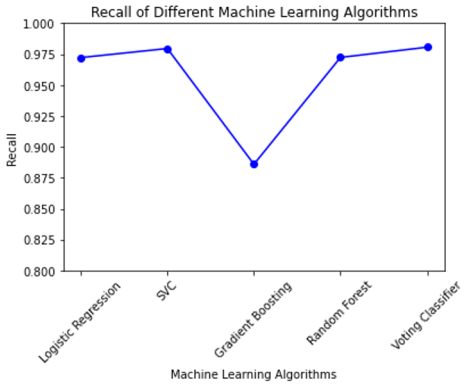

Fig. 7. Recall

4.1.4 f1 score

The f1 score, which combines Precision and Recall, measures the overall A machine learning model's performance. It is calculated by harmonically averaging precision and recall, giving each statistic equal weight.

The graph **Fig. 8.** represents the f1 score of the five machine learning algorithms. It can be observed that the **Voting classifier** has the highest f1 score.

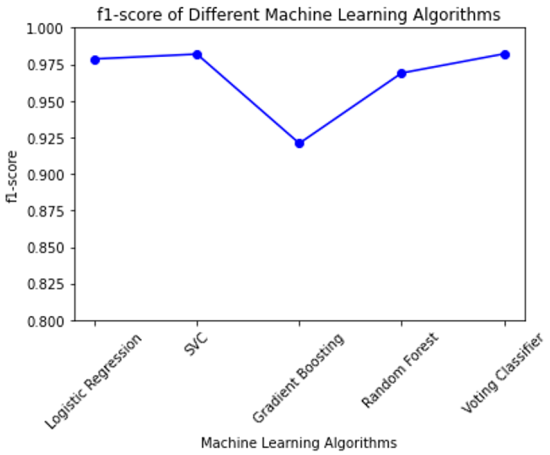

Fig. 8. f1-score

4.1.5 Confusion Matrix

It is a table used to evaluate the efficiency of a classification model. It displays how many occurrences were successfully identified by the model and how many were wrongly classified. There are four quadrants in the confusion matrix, namely True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

Graph **Fig. 9.** represents the bar groups that contain True Positive, False Positive, True Negative, and False Negative Values of the five machine-learning models.

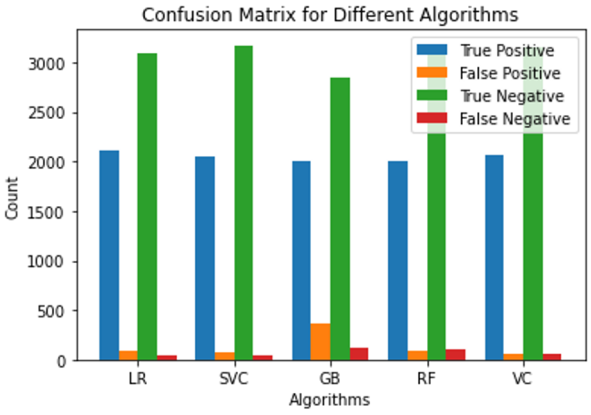

Fig. 9. Confusion Matrix

It can be concluded that the **Voting Classifier** is efficient among all the five machine-learning classifiers.

4.2 Method-2: Word2Vec:-

The data tokenized and converted into vectors using Word2Vec word-embedding technique, is then fitted into the machine learning models. Then the trained model is tested to obtain the following metrics were obtained as follows:

4.2.1 Accuracy

The graph **Fig. 10.** represents the Accuracy of the five machine-learning algorithms. It can be observed that the **Voting Classifier** has the highest Accuracy.

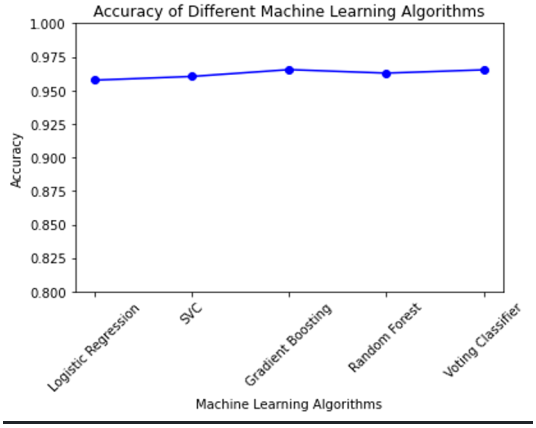

Fig. 10. Accuracy

4.2.2 Precision

The graph **Fig. 11.** represents the Precision of the five machine-learning algorithms. It can be observed that the **Voting Classifier** has the highest Precision.

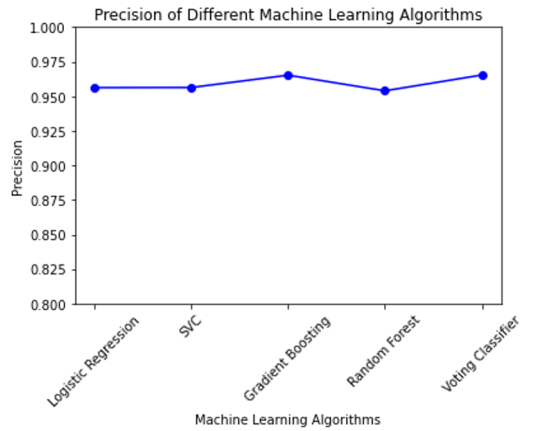

Fig. 11. Precision

4.2.3 Recall

Graph **Fig. 12.** represents the Recall of the five machine-learning algorithms.
It can be observed that the **Random Forest Classifier** has the highest Recall.

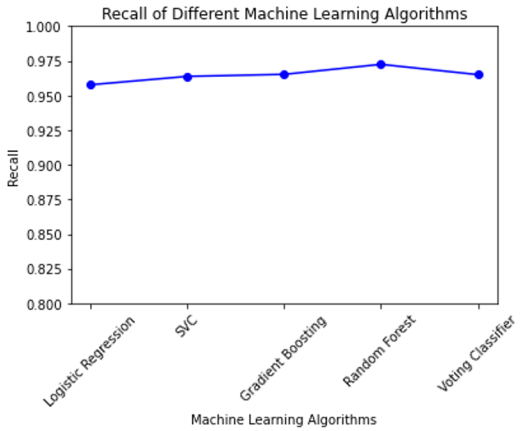

Fig. 12. Recall

4.2.4 f1 score

The graph **Fig. 13.** represents the f1 score of the five machine-learning algorithms.
It can be observed that the **Voting classifier** has the highest f1 score.

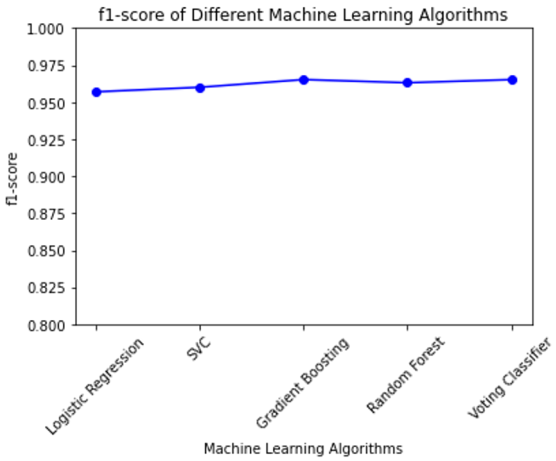

Fig. 13. f1-score

4.2.5 Confusion Matrix

The graph **Fig. 14.** represents the bar groups that contain True Positive, False Positive, True Negative, and False Negative Values of the five machine-learning models.

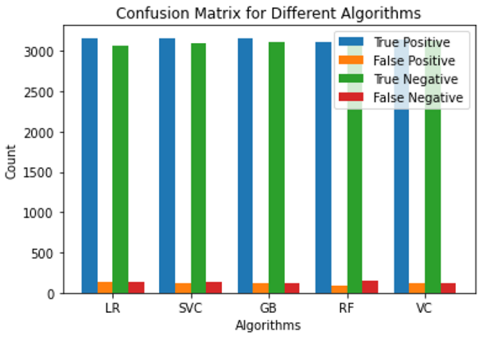

Fig. 14. Confusion Matrix

It can be concluded that the **Voting Classifier** is efficient among all the five machine-learning classifiers. Whether TF-IDF or Word2Vec word-embedding, the Voting Classifier algorithms perform well in Accuracy, Precision, Recall, and F1-Score. Models with high Accuracy and Precision better classify clickbait and non-clickbait headlines.

5 Conclusion

The primary purpose of this project was to develop a system that detects whether an article headline is clickbait. Collecting the headlines was the initial step in the process. Then, text pre-processing and word vectorization using TF-IDF and word2vec were done. The machine-learning models such as Logistic Regression, SVC, Random Forest Classifier, Gradient Boosting Classifier, and Voting Classifier were trained on the data obtained. All the models had metrics like Accuracy, Precision, Recall, and f1 score that were greater than 90%.

In our daily lives, natural language processing (NLP) and machine learning (ML) approaches combined have proven to be very much valuable. Clickbait, often misleading, is designed to grab one's attention and persuade one to click on articles or links, wasting one's time and often providing little substance. By detecting clickbait headlines, one can save valuable time and avoid falling into the trap of empty content

6 Future Scope

The project can be improved in the future in many ways, such as:

1. Constantly upgrading the machine learning model by fine-tuning its hyperparameters, experimenting with alternative algorithms, or attempting more advanced techniques such as deep learning architectures.
2. Extending the system to detect clickbait in languages other than English through multilingual NLP approaches and language-specific pre-processing processes.
3. Developing a real-time clickbait detection system capable of classifying headlines as clickbait or non-clickbait as they are published, allowing for the immediate identification and filtering of clickbait content.
4. Creating a user-friendly online application or API that allows users to enter headlines and instantly determine whether or not they are clickbait.

5. Giving users the ability to provide feedback on categorization findings, such as reporting misclassified headlines. The feedback can be utilized to train and develop the model more over time.

References

1. Smith, A., Johnson, B., Brown, C., "Clickbait Detection using Deep Learning Techniques," in International Journal of Artificial Intelligence (2021)
2. Lee, D., Kim, E., Park, S., "Feature Engineering for Clickbait Detection in Online News Headlines," in Journal of Information Science (2022)
3. Chen, H., Wang, L., Zhang, J., "A Hybrid Approach for Clickbait Detection in Social Media Headlines," in IEEE Transactions on Computational Social Systems (2023)
4. Gupta, R., Aggarwal, S., Singh, P., "Clickbait Detection using Ensemble Learning and Domain Knowledge," in Expert Systems with Applications (2023)
5. Zhang, Y., Liu, S., Wang, X., "Exploring Deep Neural Networks for Clickbait Detection: A Comparative Study," in ACM Transactions on Information Systems (2023)
6. Authors: Wang, J., Li, Y., Zhang, Q., "Clickbait Detection using Graph-based Text Representation," in IEEE Transactions on Knowledge and Data Engineering (2023)
7. Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, and Niloy Ganguly, "Stop Clickbait: Detecting and Preventing Clickbaits in Online News Media," in Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), San Fransisco, US (August 2016)
8. https://tse4.mm.bing.net/th?id=OIP.KGDS0XWdEKvFE7ufnZHUQgHaDt&pid=Api&P=0&w=300&h=300
9. https://logodownload.org/wp-content/uploads/2019/10/python-logo-0.png
10. https://scikit-learn.org/stable/install.html
11. http://nltk.org/api/nltk.corpus.html
12. https://www.youtube.com/playlist?list=PLZoTAELRMXVNNrHSKv36Lr3_156yCo6Nn
13. Avvari, Pavithra, et al. "An Efficient Novel Approach for Detection of Handwritten Numericals Using Machine Learning Paradigms." Advanced Informatics for Computing Research: 5th International Conference, ICAICR 2021, Gurugram, India, December 18–19, 2021, Revised Selected Papers. Cham: Springer International Publishing, 2022.
14. Y Jeevan Nagendra Kumar, V Spandana, VS Vaishnavi, K Neha, VGRR Devi, "Supervised Machine Learning approach for Crop Prediction in Agriculture Sector", IEEE - 5th International Conference on Communication and Electronics Systems (ICCES), ISBN: 978-1-7281-5370-4 pg: 736-741
15. Prasanna Lakshmi, K., Reddy, C.R.K. A survey on different trends in Data Streams (2010) ICNIT 2010 - 2010 International Conference on Networking and Information Technology, art. no. 5508473, pp. 451-455.
16. Jeevan Nagendra Kumar, Y., Spandana, V., Vaishnavi, V.S., Neha, K., Devi, V.G.R.R. Supervised machine learning Approach for crop yield prediction in agriculture sector (2020) Proceedings of the 5th International Conference on Communication and Electronics Systems, ICCES 2020, art. no. 09137868, pp.736- 741.

17. Sankara Babu, B., Suneetha, A., Charles Babu, G., Jeevan Nagendra Kumar, Y., Karuna, G. Medical disease prediction using grey wolf optimization and auto encoder based recurrent neural network (2018) Periodicals of Engineering and Natural Sciences, 6 (1), pp. 229-240.
18. Nagaraja, A., Boregowda, U., Khatatneh, K., Vangipuram, R., Nuvvusetty, R., Sravan Kiran, V. Similarity Based Feature Transformation for Network Anomaly Detection (2020) IEEE Access, 8, art. no. 9006824, pp. 39184-39196.
19. Y. Sri Lalitha, G. V. Reddy, K. Swapnika, R. Akunuri and H. K. Jahagirdar, "Analysis of Customer Reviews using Deep Neural Network," 2022 International Conference on Artificial Intelligence Trends and Pattern Recognition (ICAITPR), Hyderabad, India, 2022, pp. 1-5, doi:10.1109/ICAITPR51569.2022.9844183. EISBN: 978-1-6654-2521-6.
20. Sri Lalitha Y., Prashanthi G., Sravani Puranam, Sheethal Reddy Vemula, Preethi Doulathbaji and Anusha Bellamkonda, "Natural Language to SQL: Automated Query Formation Using NLP Techniques", E3S Web of Conferences Volume 391, 2023.